ePRIVO: an enhanced PRIvacy-preserVing opportunistic routing protocol for vehicular delay-tolerant networks

Article (Accepted Version)

ePRIVO: An enhanced PRIvacy-preserVing Opportunistic routing protocol for Vehicular Delay-Tolerant Networks

Naercio Magaia, Carlos Borrego, Paulo Pereira, *Senior Member, IEEE,* and Miguel Correia, *Senior Member, IEEE*

***Abstract*—This article proposes an enhanced PRIvacy preserVing Opportunistic routing protocol (ePRIVO) for Vehicular Delay-Tolerant Networks (VDTN). ePRIVO models a VDTN as a time-varying neighboring graph where edges correspond to neighboring relationship between pairs of vehicles. It addresses the problem of vehicles taking routing decision meanwhile keeping their information private, i.e, vehicles compute their similarity and/or compare their routing metrics in a private manner using the Paillier homomorphic encryption scheme.**

The effectiveness of ePRIVO is supported through extensive simulations with synthetic mobility models and a real mobility trace. Simulation results show that ePRIVO presents on average very low cryptographic costs in most scenarios. Additionally, ePRIVO presents on average gains of approximately 29% and 238% in terms of delivery ratio for the real and synthetic scenarios considered compared to other privacy-preserving routing protocols.

***Index Terms*—Privacy, Routing, Vehicular Delay-Tolerant Networks, Betweenness centrality, Similarity.**

I. Introduction

THE future network infrastructure for vehicular environments will increase the pervasiveness of the Internet and the overall connectivity by integrating every object and forming an intelligent vehicular transportation system (ITS). Examples of vehicular applications include automatic collision warning, remote vehicle diagnostics, emergency management and assistance for safe driving, vehicle tracking, automobile high-speed Internet access, and multimedia content sharing. Nevertheless, the volume of data required for such applications will continue to increase as the number of connected vehicles increases and the use cases evolve. Communication protocols for information transmission between vehicles and roadside unit (RSU) infrastructure equipment, known as vehicle-to-infrastructure (V2I), between vehicles and pedestrians, known as vehicle-to-pedestrian (V2P) as well as between vehicles, known as vehicle-to-vehicle (V2V), become more inevitable for applications of mobile content dissemination.

The opportunistic contacts enabled by vehicle-to-everything (V2X) communications are capable of providing high bandwidth communication capacity for data transmission, which forms the basis of Vehicular Delay-Tolerant Networks (VDTNs) [1]. VDTN network functions depend on the principle of cooperation between vehicles, which includes the strategies for signaling and reservation of resources (e.g., storage and bandwidth) [2]. The signaling functionality allows for other vehicles discovery and resource's reservation. When two vehicles are in communication range, they may exchange signaling information such as node type, geographical location, current path, and velocity, energy and buffer status, link rate and transmission ranges, that is considered dynamic network information. However, static network information may also be used [3]. Through social network analysis, static network information, which is more stable over time, can be leveraged and used by VDTN routing protocols to facilitate the forwarding of messages. Centrality [4], which is widely used in graph theory and network analysis, is a quantitative measure of the structural importance of a certain vehicle in relation to others within the vehicular network. In VDTNs, central nodes may be considered good candidates to be relay nodes. Among the centrality metrics, betweenness centrality [4] can be considered the most prominent, as it measures how well a node can facilitate communication among others by summing up the fraction of shortest paths between other pairs of nodes passing through it. Similarity [4], a measure of common features of a group of nodes, can be computed, for example, by finding common neighbor nodes they might have.

Computing routing metrics such as betweenness centrality or similarity requires the exchange of information between nodes. VDTN nodes represent vehicles or the entities owning and managing them, and edges the relationship between two entities. In VDTNs, information about entities owning and managing VDTN nodes and their relationships may be private. Other private information, which is also useful to improve routing performance, is the node's localization information (e.g., geographical location, current path, and velocity) because it allows predicting the period of time during which nodes' links will be in range with each other. Therefore, these links could be configured to be active only during these times which would allow saving energy that is important to network nodes with limited resources. VDTN applications would benefit from mechanisms that enforce the entities'

N. Magaia, P. Pereira and M. Correia are with INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Rua Alves Redol, nº 9, 1000-029 Lisboa, Portugal. (phone: +351-213100286; fax: +351-213145843; e-mails: naercio.magaia@tecnico.ulisboa.pt, prbp@inesc.pt, miguel.p.correia@tecnico.ulisboa.pt).

N. Magaia is also with the Department of Engineering and Design, University of Sussex, Falmer, United Kingdom.

C. Borrego is with the Department of Information and Communications Engineering, Autonomous University of Barcelona, Barcelona, Spain (email: cborrego@deic.uab.cat).

identities and/or relationship anonymity due to the sensitive or confidential nature of the entities' identities and their behaviors. A VDTN node may disclose private information that may be necessary for routing purposes, by sending private data to other nodes. Privacy-preservation techniques allow protecting privacy through masking, modification and/or generalization of the original data without sacrificing the data utility [5].

If it is considered that some vehicles might misbehave maliciously or not by, for example, not forwarding others' vehicles information, private information such as contacts' history, list of neighbors, etc., which is required for computing some routing metrics should not be disclosed to misbehaving vehicles [6]. However, vehicles should be able to use part of this information, if necessary. Furthermore, despite the good routing performance of some of the proposed routing protocols [3], [4], most of the security issues presented in [6] (such as confidentiality, integrity, privacy, etc.) were not considered. For instance, to deal with confidentiality and privacy, vehicular communication protocols should implement cryptographic mechanisms.

This article proposes an enhanced PRIvacy-preserVing Opportunistic routing protocol for Vehicular Delay-Tolerant Networks (ePRIVO). ePRIVO builds upon our previous work [5]. It models a VDTN as a time-varying neighboring graph where edges correspond to the neighboring relationship between pairs of vehicles. It addresses the problem of vehicles taking routing decisions meanwhile keeping their information private, i.e, vehicles compute their similarity and/or compare their routing metrics in a private manner using the Paillier homomorphic encryption scheme.

The contributions of this article are summarized as follows:

- ePRIVO, an enhanced and efficient privacy-preserving routing protocol for VDTNs is proposed. A detailed threat model and privacy analysis are presented and different forwarding policies are considered to enhance its performance. Addressing privacy issues in vehicular networks is a hot research topic;
- Two anonymization methods (i.e., binary anonymization and neighborhood randomization) to ensure privacy are defined. They are used by VDTN nodes to exchange neighborhood information;
- A secure approach to determine the similarity between two interacting nodes is proposed. It allows determining the least possible number of nodes to be shared between the two interacting nodes without disclosing overtime their nodes' degrees.
- A privacy mechanism that uses the Paillier homomorphic encryption scheme is proposed. It enables nodes to compare their routing metrics without disclosing them.
- Through extensive simulations with synthetic mobility models and a real-trace, and even by considering much stronger hence nowadays more secure keys, it was shown that our privacy-preserving approach achieves good performance in vehicular networking environments.

The remainder of this article is organized as follows. Section II presents related work. Section III introduces notations and assumptions. It also introduces relevant social metrics and cryptographic mechanisms for this work. Section IV presents the ePRIVO protocol. Section V presents the privacy analysis. In Section VI, the simulation model and results are presented. Finally, Section VII presents concluding remarks and future work.

II. Related Work

According to the literature [7], privacy breaches can be classified as identity disclosure, link disclosure, and attribute disclosure. Identity disclosure is the case when the identity of the individual associated with the node is revealed. Link disclosure happens when the sensitive relationship between the individuals is disclosed. Attribute disclosure is the case when the sensitive data associated with the node, owned by an individual, is compromised. Moreover, there are several types of sensitive information such as node attributes, specific link relationships between nodes, nodes degrees, neighborhoods of some target nodes, etc.

Anonymization methods [7] can be used to protect privacy if sensitive information needs to be processed elsewhere. There are three main anonymization methods, namely: (*i*) k-anonymity privacy preservation via edge modification, that modifies graph structure by successive deletions and additions of edges so that each node in the modified graph is indistinguishable with at least $k-1$ other nodes in terms of a given network property; (*ii*) edge randomization, that modifies the graph structure by randomly adding/deleting edges or by switching edges; and (*iii*) cluster-based generalization, where nodes and edges are clustered into groups and anonymized into a super-node.

It is commonly assumed in Wireless Ad Hoc Networks that nodes are willing to share their private information for the sake of the network's performance. For instance, some routing protocols that address privacy issues in DTNs and VDTNs are referenced next. Routing approaches such as [8], [9], [10] ensure attribute privacy. The location used by the source node to send messages is protected in [8]. The context, e.g. personal information, residence, work, hobbies, interest profiles, etc., which is used for forwarding is protected in [9], [10]. In [11], the location information of individual vehicles during its communication with RSUs is protected. However, privacy issues resulting from V2V communications are not considered. Both location and identity privacy are ensured through a group communication scheme, i.e., a cluster-based generalization, for sparse VDTNs in [12]. Privacy can only be guaranteed once groups are formed, and forming them may take some time.

An approach based on Privacy by Architecture, in which minimal information that identifies a user is sent to a Certificate Authority when requesting a certificate, was proposed in [13]. It requires a complex pseudonym-based cryptographic key management depending on the number of messages to be sent. An identity-hidden message indexing mechanism is used to protect the receiver's location in [14]. It enables the receiver to query for messages whose destination is itself without revealing its identity to socialspot RSUs (e.g., intersections, famous shopping malls, movie theatres) in a VDTN. In [15], an adaptive mechanism for achieving user anonymity that ensures identity privacy is proposed. Identity privacy can be

compromised if an attacker combines external knowledge with observed network structure [7].

In [16], an approach that ensures link privacy has been proposed where instead of transmitting the list of friends of the sender as a list of nodes, a modified and obfuscated one is transmitted. Two strategies, namely meeting relationship anonymity and forwarder anonymity, were proposed in [17] to protect private information (e.g., probabilities of meeting other nodes) in utility-based routing protocols. The cryptographic mechanisms used leaks information such as the order of the plaintexts, besides the results of the ciphertexts being known by the intervening parties. A privacy-preserving packet forwarding protocol, which is based on threshold credit-based incentive mechanism, is proposed in [18]. It addresses the resisting layer-adding attack by outsourcing the privacy-preserving aggregated transmission evidence generation for multiple resource-constrained vehicles to the cloud side. A trusted third party (TTP) is required to generate evidence of the vehicles' behavior, and attribute privacy is only considered at the cloud-assisted VDTN. In [19], a privacy-preserving prediction-based probabilistic routing was proposed to avoid disclosing the mobility patterns of the nodes. Messages are forwarded comparing aggregated information about communities instead of individual nodes. More efficient privacy-preserving routing protocols such as ePRIVO could be used to disseminate messages within a community.

Other privacy techniques have been proposed in the literature. For instance, with homomorphic encryption – proposed by Rivest et al. in 1978 [20] – a node can carry out computations on encrypted values, without needing to decrypt them first. In [21] and PrivHab+ [22], privacy-preserving routing protocols based on additive homomorphic encryption (Paillier cryptosystem [23]) were proposed. The former, which was proposed for peer-to-peer networks, allowed a node to calculate its similarity to other nodes using multivariate polynomial evaluation, meanwhile, the latter, which was proposed as a secure geographical routing protocol for DTNs, allowed nodes to compare their habitats in order to choose the best forwarder for every message, respectively. Besides PrivHab+, which is not suitable for social DTNs and only ensures attribute privacy, none of the above approaches protects the nodes' private information if it has to be shared and processed elsewhere (link privacy), or used during routing decisions (attribute privacy). ePRIVO ensures both link and attribute privacy.

In [5], a previous work of ours, a privacy-preserving routing protocol, known as PRIVO, was proposed for DTNs. The approach used by PRIVO's anonymization methods to determine the least possible number of nodes that two nodes should share could disclose over some interactions the nodes' degrees. PRIVO also lacked a privacy analysis of its mechanisms and the performance evaluation was done with key sizes considered unsecure nowadays. On the other hand, ePRIVO is an efficient and enhanced privacy-preserving routing protocol for VDTNs. In VDTNs, differently from DTNs, vehicles use V2X communications to disseminate messages. ePRIVO proposes a secure mechanism to allow nodes to determine their similarity, therefore avoiding to disclose the nodes' degrees after some interactions. A detailed threat model and privacy analysis of ePRIVO are presented. The performance evaluation of ePRIVO was done with much stronger, thus nowadays more secure key sizes. And last, but not least important, different forwarding policies were considered to enhance ePRIVO's performance.

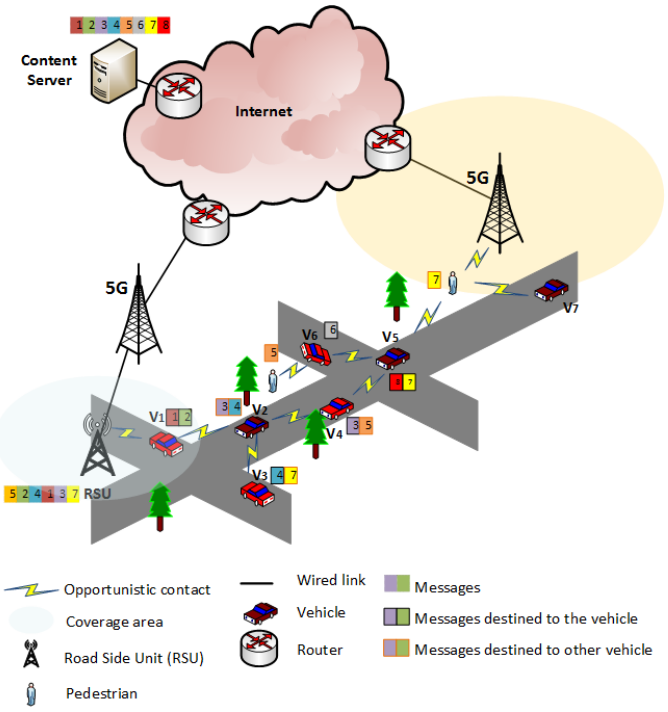


Fig. 1. Illustration of a vehicular application integrating 5G network and V2X opportunistic communications.

III. Preliminaries and Background

A. Assumptions and notations

a) Notation: A notation similar to [4] is used. A VDTN neighboring graph is modeled as a time-varying graph $\mathcal{G} = (V, E, \mathcal{T}, w)$ where each vertex $v \in V$ corresponds to a node in the network and each edge $e = (i, j) \in E$ represents the relationship between these nodes (i.e., that these nodes have encountered before). The relations among nodes are assumed to take place over a time span $\mathcal{T} \in \mathbb{T}$ known as the lifetime of the network; $w : E \times \mathcal{T} \rightarrow [0, 1]$ is called *weight* function and indicates the strength of an edge at a given time.

Let a footprint of $\mathcal{G}$ from t_1 to t_2 be defined as a static graph $G^{[t_1,t_2)} = (V, E^{[t_1,t_2)})$ such that $\forall e \in E, e \in E^{[t_1,t_2)} \iff \exists\ t \in [t_1, t_2), \quad w(e, t) \in [0, 1]$, i.e., the footprint aggregates all interactions of a given time window (or timeslot) into static graphs. Let $\tau = [t_0, t_1), [t_1, t_2), \ldots, [t_i, t_{i+1}), \ldots$ (where $[t_k, t_{k+1})$ can be noted τ_k) be the lifetime $\mathcal{T}$ of the time-varying graph partitioned in sub-intervals. The sequence $\mathrm{SF}(\tau) = G^{\tau_0}, G^{\tau_1}, \ldots$ is called sequence of footprints of $\mathcal{G}$ according to τ.

b) Scenario: Consider the network topology in Figure 1, where vehicles and pedestrians, travel around the city and the deployed RSUs provide coverage over a certain area. RSUs are placed at the intersections similarly to what is done by current optimal placement algorithms [24]. RSUs are

connected through wired links to 5G Radio Access Networks (RANs) that are also connected to the content server on the Internet. Vehicles requiring mobile data (or messages) send their requests to the content server via V2X communication links. The requested data is delivered from the content server to the 5G RAN and from the 5G RAN to the RSUs or pedestrians via the wired and wireless links, respectively. It is assumed that the wired links provide relatively high bandwidth hence ensuring that the requested data is delivered to RSUs prior to the delay-tolerant dissemination between RSUs and vehicles. RSUs and pedestrians will further disseminate the data to the users in the vehicles that requested it through opportunistic communication, i.e., in a store-carry-and-forward manner, that occurs when the vehicle moves into the communication coverage of RSUs or pedestrians.

In Figure 1, some vehicles such as V_1, V_5 and V_6 only carry messages destined to themselves, conversely to V_2 that does not carry any messages destined to itself. If V_2 requests message 8, and since V_5 has it, V_2 would have to rely on V2V communications between V_5 and V_4, and between V_4 and itself to get the message. Alternatively, it could also wait until it gets into communication range of the RSU to get message 8 through a V2I communication with the RSU. Please note that the words vehicle and node are used interchangeably throughout the article.

c) Node capability: Each node has a Unique IDentifier (UID) that cannot be spoofed. Upon an encounter between two nodes, a secure communication channel between the two is used through cryptographic mechanisms that ensure confidentiality.

d) Attack model: It is assumed an adversary model in which malicious nodes are also vehicles similarly to good nodes. However, they try to learn the private information of others through the following brute-force attacks (BFAs):

- Neighboring graph BFA: In this attack, a malicious node tries to discover other nodes' historical encounter information (i.e., the neighboring graph) by querying for all possible nodes and storing the received weights.
- Node metric BFA: In this attack, a malicious node tries to discover the routing metric of the other node it came in contact with by sending its different (fake) routing metrics over a series of contacts until the correct routing metric value of the other node is found.

B. Background

1) Social metrics: A variety of network information has been used to address the challenging task of finding the most suitable node to forward messages in a VDTN, namely dynamic network information (e.g., location, traffic, encounter information, etc.) and social network information (e.g., social relations among nodes). However, social network information is more stable over time than dynamic network information and can be leveraged by VDTN routing protocols to facilitate the forwarding of messages [3].

a) Ego betweenness centrality: Centrality of a node in a network is a quantitative measure of the structural importance of this node in relation to others within the network. Typically, a node can be considered as central if it plays an important role in the network's connectivity, for example, if it is more apt to connect to others in the network. The three most common centrality metrics are degree, closeness and betweenness centrality [4]. Degree centrality is defined as the number of links (that is, direct neighbors) incident upon a given node. Besides being a local metric, it only takes into account the number of neighbors of a given node, thus not taking into consideration the global structure of the network. Closeness centrality is defined as the total shortest path distance from a given node to all other nodes. However, it lacks applicability in networks with disconnected components. Betweenness centrality is defined as the number of geodesics (shortest paths) passing through a given node. Betweenness centrality can be perceived as a measure of the load placed on a given node since it measures how well a node can facilitate communication among others. Betwenesss takes into account the global structure of the network, and it can be applied to networks with disconnected components. ePRIVO uses a betweenness centrality metric that does not require global knowledge, hence being more suitable for VDTNs.

An ego network [25] (also known as the neighborhood network of the ego) is defined as a network that consists of a central node (ego) along with its direct neighbors (the other nodes the ego is directly connected to) and all links among these neighbors. The shortest paths, due to the structure of the ego network, are either of length 1 or 2. Every single pair of non-adjacent direct neighbors must have a shortest path of length 2 which passes through the ego. Shortest paths of length 1 do not contribute to the betweenness centrality computation. If $\mathcal{A}$ is an adjacency matrix of graph $\mathcal{G}$, then $\mathcal{A}^2_{i,j}$ contains the number of geodesics of length 2 connecting vertices i and j. The number of shortest paths between i and j is given by $\mathcal{A}^2[\mathbf{1} - \mathcal{A}]_{i,j}$ (where $\mathbf{1}$ is a matrix of all 1's).

The ego betweenness centrality (c_{EBC}) is the sum of the halved reciprocal entries $\mathcal{A}^2[\mathbf{1} - \mathcal{A}]_{i,j}$ such that $\mathcal{A}_{\mathbf{i},\mathbf{j}} = \mathbf{0}$.

b) Similarity: Similarity [26] expresses the number of common features of a group in social networks. In sociology, the probability of two individuals being acquainted increases with the number of common acquaintances between them [27]. In computer networks, the similarity between nodes i and j can be defined as the number of common neighbors among them. Therefore, the more common neighbors they have, the more similar they are.

2) Homomorphic encryption: In cryptography, finding common elements in two private sets without exposing the sets themselves is known as the Private Set Intersection (PSI) problem [28]. For instance, an algorithm that solves the PSI problem would allow a trusted node to send an encrypted version of some data to be processed by an untrusted node and the latter would perform computations on this encrypted data without knowing anything of the data's real value and send back the result. The trusted node would expect the decrypted result to be equal to the intended computed value as if it was performed on the original data. For example, with homomorphic encryption, a node can carry out computations on encrypted values, without decrypting them first.

If addition operators are considered, the scheme is addi-

tively homomorphic. Likewise, if multiplication operators are considered, the scheme is multiplicatively homomorphic. An additive homomorphic encryption scheme is the one in which two numbers encrypted with the same key $\mathcal{E}(a)$ and $\mathcal{E}(b)$ can be added without being first decrypted, i.e., one can efficiently compute $\mathcal{E}(a+b)$ without decrypting them.

In the Paillier cryptosystem [29], which is an additive homomorphic encryption scheme, when entity i wants to send message m to entity j, entity i selects random primes p and q and constructs $n = pq$; plaintext messages are elements of $\mathbb{Z}_n$ and cyphertext are elements of $\mathbb{Z}_{n^2}$. Entity i picks a random $g \in \mathbb{Z}^*_{n^2}$ and verifies that $\exists \mu$ where $\mu = (L(g^\lambda \bmod n^2))^{-1} \bmod \text{n}$, $L(x) = (x-1)/n$ and $\lambda = \text{lcm}(p-1, q-1)$. If $\nexists \mu$ then a new random $g \in \mathbb{Z}^*_{n^2}$ must be picked. Entity i's public key p_k is (n, g) and private key s_k is (λ, μ).

To encrypt a message m, entity j picks a random $r \in \mathbb{Z}^*_n$ and computes the ciphertext $c = \mathcal{E}(m) = g^m \cdot r^n \bmod n^2$, therefore cyphering with p_k. To decrypt c, entity i computes $\mathcal{D}(c) = \left(L(c^\lambda \bmod n^2)\right)^{-1} \cdot \mu \bmod n = m$, therefore deciphering with s_k.

Let $\mathcal{E}(a) = g^a \cdot r_1^n \bmod n^2$ and $\mathcal{E}(b) = g^b \cdot r_2^n \bmod n^2$. Entity j can compute the sum this way: $\mathcal{E}(a+b) = \mathcal{E}(a) \cdot \mathcal{E}(b) \bmod n^2 = g^{a+b} \cdot (r_1 r_2)^n \bmod n^2$.

Let $\mathcal{E}(a) = g^a \cdot r_1^n \bmod n^2$ and k be a non-encrypted constant. Entity j can compute the multiplication by a non-encrypted constant $\mathcal{E}(k \cdot a) = \mathcal{E}(a)^k \bmod n^2 = g^{k \cdot a} \cdot (r_1)^n \bmod n^2$.

IV. The ePRIVO protocol

This section introduces the enhanced PRIvacy-preserVing Opportunistic routing protocol for Vehicular Delay-Tolerant Networks (ePRIVO). ePRIVO detects and utilizes the inherent social network structure to facilitate message forwarding in VDTNs. It models a VDTN as a time-varying neighboring graph where vertices correspond to nodes and edges correspond to the neighboring relationship between pairs of nodes.

ePRIVO ensures privacy by means of anonymization and homomorphic encryption. It uses anonymization and homomorphic encryption to avoid disclosing historical information associated with each node's neighboring graph. When two nodes meet, they do not share private information associated with their routing metrics, which is necessary to identify the best message forwarder. Nodes compare these metrics in a private manner also using homomorphic encryption.

The ePRIVO protocol is composed of the following steps: construction and anonymization of the neighboring graph, determination of routing metrics and the routing algorithm.

In order to facilitate future references, frequently used notations in this article are listed in Table I.

TABLE I
The most used notations in this article

Notation	Meaning
$x_{i,j}(t)$	The separation period between nodes i and j
τ_k	The elapsed time (or timeslot) between t_k and t_{k+1}
$\delta_{(i,j),\tau_k}(x)$	The average separation period between nodes i and j at timeslot τ_k
$w_{i,j}$	The ePRIVO weight
ρ	The anonymization threshold
ε	The weight threshold
η	The number of timeslots
ζ_{AB}	The similarity between nodes A and B
H_v	Neighboring graph of node v

A. Construction of the neighboring graph

Let $H = (V_H, E_H)$ be a subgraph of $G = (V, E)$, denoted $H \subset G$, if and only if $V_H \subset V$ and $E_H \subset E$. H is a local subgraph (hereafter *neighboring graph*) with respect to a vertex $v \in V$, if and only if all vertices in the subgraph can be directly reached from v. Let $x_{i,j}(t)$ denote the separation period between nodes i and j, τ_k denote the elapsed time and $n_{i,j}$ be the number of times that nodes i and j were away from each other. So, $x_{i,j}(t) = 0$ means that nodes i and j are within communication range at time $t \in \tau_k$, otherwise $x_{i,j}(t) = 1$. The time-varying average separation period between nodes i and j at timeslot τ_k (hereafter average separation period) is given by

$$\delta_{(i,j),\tau_k}(x) = \frac{\int_{\tau_k} x_{i,j}(t)\, dt}{n_{i,j}} \tag{1}$$

The normalized average separation period $\hat{\delta}_{\tau_k}$ at timeslot τ_k is given by

$$\hat{\delta}_{\tau_k} := \hat{\delta}_{(i,j),\tau_k}(x) = 1 - \frac{\delta_{(i,j),\tau_k}(x)}{|\tau_k|} \tag{2}$$

where $|\tau_k|$ is the duration of τ_k. Here, the lifetime $\mathcal{T}$ consists of many days, and each day consists of a fixed number of timeslots. The average separation period aims at capturing the evolution of social interactions in similar timeslots.

The normalized average separation period in the same timeslot τ_k over consecutive days is updated using an exponential weighted moving average (EWMA) as follows

$$\Delta^{\mathfrak{t}}_{\tau_k} = \begin{cases} \hat{\delta}^{\mathfrak{1}}_{\tau_k}, & \mathfrak{t} = 1 \\ (1-\alpha) \cdot \Delta^{\mathfrak{t}-1}_{\tau_k} + \alpha \cdot \hat{\delta}^{\mathfrak{t}}_{\tau_k}, & \mathfrak{t} > 1 \end{cases} \tag{3}$$

where α is the smoothing factor, and $0 < \alpha < 1$. $\hat{\delta}^{\mathfrak{t}}_{\tau_k}$ is the value of $\hat{\delta}_{\tau_k}$ at day $\mathfrak{t}$ and $\Delta^{\mathfrak{t}}_{\tau_k}$ is the estimate of $\hat{\delta}_{\tau_k}$ at any day $\mathfrak{t}$. If not updated, i.e., $\hat{\delta}^{\mathfrak{t}}_{\tau_k} = 0$ which means that nodes i and j were away from each other at τ_k of day $\mathfrak{t}$, it is depreciated as follows

$$\Delta^{\mathfrak{t}}_{\tau_k} = (1-\alpha) \cdot \Delta^{\mathfrak{t}-1}_{\tau_k} \tag{4}$$

The unbiased variance estimator $\hat{\sigma}$ at timeslot τ_k of day $\mathfrak{t}$ ($\hat{\sigma}^{\mathfrak{t}}_{\tau_k}$), which allows measuring the variability of $\hat{\delta}_{\tau_k}$, is given by

$$\hat{\sigma}^{\mathfrak{t}}_{\tau_k} = \begin{cases} \left|\Delta^{\mathfrak{1}}_{\tau_k} - \hat{\delta}^{\mathfrak{1}}_{\tau_k}\right|, & \mathfrak{t} = 1 \\ (1-\beta) \cdot \hat{\sigma}^{\mathfrak{t}-1}_{\tau_k} + \beta \cdot \left|\Delta^{\mathfrak{t}}_{\tau_k} - \hat{\delta}^{\mathfrak{t}}_{\tau_k}\right|, & \mathfrak{t} > 1 \end{cases} \tag{5}$$

where $0 < \beta < 1$.

The social strength among nodes in a specific daily timeslot may provide insights on their social strength in consecutive timeslots on the same day, therefore increasing the probability of nodes being capable of transmitting data as transmissions could be resumed, with high probability, on the same timeslot on the next day. The time-varying ePRIVO weight $w_{i,j}$ (hereafter *pweight*) over a daily time-period is given by

$$w_{i,j} = \frac{1}{|\tau^{\mathfrak{t}}|} \sum_{k=1}^{|\tau^{\mathfrak{t}}|} \Delta_{\tau_k}^{\mathfrak{t}} \tag{6}$$

where $|\tau^{\mathfrak{t}}|$ is the number of timeslots of day $\mathfrak{t}$; *pweight* shows the neighboring relationship among nodes and gives hints about the forwarding opportunities between them, i.e., larger $w_{i,j}$ indicates a better future contact probability between nodes i and j.

In ePRIVO, nodes' routines are used to quantify the time-varying strength of social ties between nodes. For instance, if daily routines are considered, each node computes the average separation periods to other nodes during the same set of daily timeslots over consecutive days.

B. Anonymization of the neighboring graph

A VDTN node may disclose private information by sending private data to other nodes. Privacy-preservation techniques allow protecting privacy through masking, modification and/or generalization of the original data without sacrificing data utility.

1) The similarity privacy mechanism: ePRIVO proposes the similarity privacy mechanism, which is based on the Paillier homomorphic encryption scheme. The mechanism addresses the PSI problem [21] by allowing a node, say A, to calculate its similarity to another node, say B, in a private manner.

Let H_A and H_B be the neighboring graphs of nodes A and B, respectively. A queries B in order to compute the number of common nodes in their neighboring graphs without disclosing the neighboring graphs themselves. The cardinality of the intersection between the two neighboring graphs, which is computed using multivariate polynomial evaluation [21], is used to find the similarity between the two nodes (ζ). Let $A \rightarrow B \ : < message >$ denote a message sent from A to B. Let p_k and s_k be public and private keys, respectively. The mechanism works as follows:

1) Node A builds a polynomial having roots in each of the nodes contained in its H_A, i.e., A computes the $n+1$ coefficients α_0, ..., α_n of the polynomial

$$f(y) = \alpha_0 + \alpha_1 y + \alpha_2 y^2 + ... + \alpha_n y^n$$

for which $f(H_{A_i}) = 0$ for any node in its neighboring graph.

2) Node A encrypts the coefficients and sends them to B.

$$A \rightarrow B \ : < \mathcal{E}_{p_{k_A}}(f(y)) >$$

3) Node B uses the homomorphic properties of the encryption scheme to evaluate the polynomial for each node in its binary vector Γ_B, and multiplies each result by a random once-use number, obtaining $\mathcal{E}_{p_{k_A}}(nonce \cdot f(H_{B_i}))$. Each node N creates a binary vector $\Gamma_N = \langle \gamma_1, ..., \gamma_M \rangle$, where $M = |H_N|$, for the nodes in its neighboring graph. If node N has node $i \subset V$ in its H_N then $\gamma_{N_i} = 1$ else $\gamma_{N_i} = 0$.
4) Node B adds all evaluated polynomials to a list and permutates the order. Then, B sends the list back to A.
5) Node A decrypts each ciphertext and counts the number of zeros that it received, after receiving the list of evaluated polynomials from node B. If there is a node in the intersection $H_A \bigcap H_B$, then the ciphertexts decrypt to zero. Otherwise, it decrypts to a random value. Hence, ζ_{AB} is the number of zeros decrypted by node A.

2) The link privacy mechanism: ePRIVO deals with link disclosure since each node's ego network contains the list of neighbors and their social strengths. To ensure link privacy, ePRIVO also uses two anonymization techniques that are suitable for VDTNs as they ensure data utility: neighborhood randomization and binary anonymization.

3) Neighborhood randomization: It consists in partially hiding each node's neighboring graph containing its historical encounter information. When two nodes are in communication range, they only exchange ζ_{AB} nodes in their neighboring graphs. If $w_{i,j}$ is high, it might mean that nodes i and j have a strong tie (i.e., that they meet often), or even that they have met recently. The latter may be a random link, that is, a recent occasional connection that looks like a strong tie.

Neighborhood randomization works as follows: upon an encounter between nodes i and j and differently from PRIVO, they run the similarity privacy mechanism to obtain ζ_{AB}. Randomly selecting nodes to add to the anonymized neighboring graph that will be shared allows mixing random contacts with strong contacts, therefore hiding the contact patterns among neighbors since *pweights* are constantly being updated. If i and j re-encounter after a short period of time, they can share the same previous information, therefore, avoiding to disclose more historical information. Ideally, upon an encounter between nodes i and j, the anonymized neighboring graph of j should only contain information of common nodes it has with i. This information is useful for i to update its ego network.

4) Binary anonymization: It consists in replacing the *pweight* associated to a given link with 1 or 0, if the weight is above or below a given anonymization threshold (ρ), respectively. This technique converts the weighted (randomized or not) neighborhood graph into an unweighted one, therefore hiding the *pweight* associated to a given edge. The selection of ρ is also limited by the utility of the neighboring graph.

Consider, for example, that node a has nodes b, c and d as its neighbors with *pweights* ($w_{a,b} = 0.05, w_{a,c} = 0.15, w_{a,d} = 0.65$). If ρ is set to 0.1, the anonymized *pweights* are ($w^*_{a,b} = 0, w^*_{a,c} = 1, w^*_{a,d} = 1$). But, if instead ρ is set to 0.25, the anonymized *pweights* would become ($w^*_{a,b} = 0, w^*_{a,c} = 0, w^*_{a,d} = 1$). If node a meets another node, say node e, a would tell e that its neighbors are ($w_{a,c} = 1, w_{a,d} = 1$) for $\rho = 0.1$ and ($w_{a,d} = 1$) for $\rho = 0.25$.

C. Determination of routing metrics

ePRIVO represents the dynamics of the social structure as time-varying weighted neighboring graphs, where the weights (i.e., social strengths among nodes) express the average separation period over different timeslots.

1) Ego betweenness centrality: Each node's ego network corresponds to its neighboring graph if the *pweigths* are above a given weight threshold (ε). Since the connections among the ego direct neighbors are also necessary for the ego network, each node shares its anonymized neighboring graph (as explained in Section IV-B) with its neighbors.

Given a set of configuration parameters (see Section VI-A for more details), the determination of ε can be seen as an optimization problem consisting in finding the ε that maximizes (or minimizes) a certain routing performance metric (e.g., finding ε that maximizes the delivery ratio).

2) Weighted similarity to the destination: Let $\mathcal{A}_n$ be the weighted adjacency matrix of node n at a given timeslot. Let $\mathcal{A}_{n_{i,j}} = w_{i,j}$. If nodes i and j have met before, then $w_{i,j} \neq 0$; otherwise, $w_{i,j}$=0. The weighted similarity of n to a destination node d (s_d) is obtained by summing the non-zero row entries in $\mathcal{A}_{n_{i,d}} | i \neq n$. If n never met d but node i belonging to n's neighboring graph did, n may infer that i is a more suitable forwarder to d than it through i's anonymized neighboring graph.

3) Mean time to encounter: Besides *pweight*, ePRIVO also uses a metric called *mean time to encounter* (MTTE) to determine the best message forwarder to a given destination taking into account the average separation period at each timeslot and the expected time necessary for the two nodes to re-encounter. Specifically, given that in ePRIVO each node keeps an estimate of the average separation period at each timeslot that is updated as nodes encounter each other, ePRIVO predicts the most probable timeslot for future contacts also taking into account the shortest time to re-encounter. As an example, consider that node a meets nodes b and c at 2pm and 5pm for 10 and 15 minutes, respectively. At 8pm, node a receives a message destined to node d that is expected to meet nodes b and c on the next day. When node a computes the average separation periods of b and c, it also considers the time to re-encounter nodes b and c in the following day assuming that these nodes maintain similar habits.

D. Routing algorithm

This section describes the routing algorithm of ePRIVO, that is, the messages exchanged using the Paillier homomorphic encryption scheme and the routing decision process.

1) The attribute privacy mechanism: ePRIVO ensures attribute privacy, as regardless of the metric ($\mathfrak{m}$) used by the routing algorithm (*pweight*, similarity to the destination, or ego betweenness centrality – $\{w_{i,j},\ s_d,\ c_{\text{EBC}}\} \subset \mathfrak{m}$), which represent utility, when two nodes meet they find the best forwarder in a private manner using the Paillier homomorphic encryption scheme.

Let A be a node carrying a set of messages $\mathcal{M}$ and node B be a neighbor of A. Let $A \rightarrow B\ : < message >$ denote a message sent from A to B. Upon an encounter, A wants to know if B is a better forwarder to carry $m \in \mathcal{M}$ destined to node D. Let p_k and s_k be public and private key, respectively.

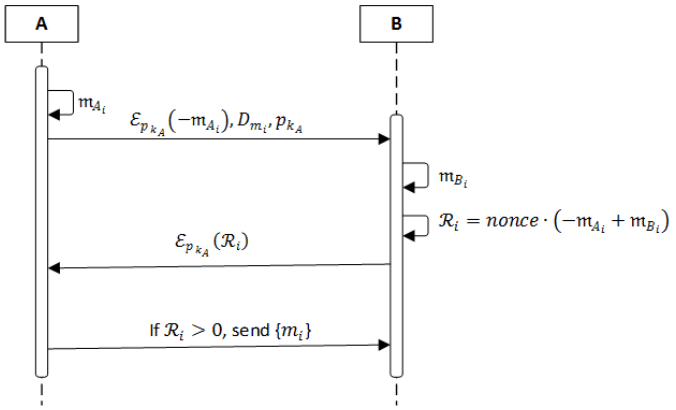


Fig. 2. Scheme of the messages exchanged during the execution of ePRIVO

The exchange of messages in ePRIVO works as follows:

1. Node A calculates metric $\mathfrak{m}$ for each $m \in \mathcal{M}$ using the information it has available.
2. Each time A establishes a contact with another node, it announces: $-\mathfrak{m}_i\ \forall\ D_i \subset m_i |\ i = 1, 2, \ \ldots |\mathcal{M}|$, the destination of the message and its public key (p_{k_A}) to B. Node A multiplies the metric $\mathfrak{m}_i$ by -1 to reduce the number of cryptographic operations to be performed by node B.
$$A \rightarrow B\ : < \mathcal{E}_{p_{k_A}}(-\mathfrak{m}_{A_i}), D_{m_i},\ p_{k_A} >$$
3. Node B performs for each metric received the following operations: first, B sums $-\mathfrak{m}_{A_i}$ to the corresponding metric $\mathfrak{m}_{B_i}$, then it multiplies the result by a random once-use number (nonce) to randomize it. Without the multiplication, A would be able to obtain $\mathfrak{m}_{B_i}$. Then B sends the result $\mathcal{R}_i = nonce \cdot (-\mathfrak{m}_{A_i} + \mathfrak{m}_{B_i})$ to A.
$$B \rightarrow A\ :\ < \mathcal{E}_{p_{k_A}}(\mathcal{R}_i) >$$
4. A decrypts the received comparisons for each m_i.
$$\mathcal{D}_{s_{k_A}}\left(\mathcal{E}_{p_{k_A}}(\mathcal{R}_i)\right)$$
Node A knows that if $\mathcal{R}_i > 0 \rightarrow \mathfrak{m}_A < \mathfrak{m}_B$ which means that node B is a better forwarder. If that is the case, A forwards m_i to B.
$$A \rightarrow B\ :< m_i >$$

Fig. 2 provides a scheme of the messages exchanged during each phase of the protocol.

Obtaining the best forwarder can be demanding in terms of CPU, energy, etc., due to the number of messages that have to be exchanged in the process. In ePRIVO, each node has a secure forwarding table (SFT) containing entries *<DestinationNode (DN), BestForwarder (BF)>* that is updated each time a node meets another one that is a better forwarder than it. When the average separation period between two nodes is updated, if one of those nodes is a BF in the SFT, the entry is removed. SFT allows reducing the number of messages exchanged when two nodes meet therefore reducing also ePRIVO's consumption of resources.

2) The routing decision process: Four variants of ePRIVO are proposed. ePrivoASP uses as routing metric *pweight*. ePrivoMTTE uses as routing metric the mean time to encounter. ePrivoSDBC, which is the social version of ePRIVO, uses as routing metric weighted similarity to the destination and ego betweenness centrality. ePrivoCOMBINED is a combination of ePrivoMTTE and ePrivoSDBC. It results from multiplying the routing metrics of ePrivoMTTE and ePrivoSDBC.

3) Forwarding policy: After the execution of the attribute privacy mechanism, node A knows if the execution was successful and if node B is more suitable to carry message m_i towards its destination. Since the number of copies of every message flowing through the network will be directly determined by the forwarding policy used, A decides if the message has to be forwarded to B, and if it will keep a copy of m_i. That is, A decides based on the forwarding policy used and this decision may have an impact on the performance of ePRIVO. ePRIVO is compatible with any forwarding policy.

Next, a not exhaustive list of forwarding policies is presented. A study and analysis aiming at identifying the best policy for each scenario will be conducted in Section VI-B4. Node A always forwards $m \in \mathcal{M}$ to another more suitable node, say node B, but in the

- *direct single-copy policy*, no copies of m are created.
- *direct multi-copy policy*, node A keeps a copy of m.
- *limited multi-copy policy*, node A can only create a limited number of copies of m. When A reaches the threshold for m, no more copies of m are created, nor it is forwarded to another node, say node C. The threshold can be defined based on different strategies for every node and message.

V. Privacy Analysis

This section analyses the knowledge obtained by each participant of ePRIVO under the scope of secure multi-party computations [30].

On a network where different participants hold each an input, a secure multi-party computation consists in computing a function on any input and ensuring that no more information is revealed to a participant (or node), say A, than what can be inferred from A's input and the computed output. Two adversary modes are considered: active and passive. In the former, node A executes the protocol and then makes inferences to obtain knowledge about the inputs of another node, say B. In the latter, node A tampers its messages to try to obtain an advantage. Then, a reasoning about the privacy obtained in the two modes is presented.

A. Passive adversary mode

To consider ePRIVO as a secure protocol [31], it is necessary to prove that it reveals only the result of the function and the inferences that can be deduced from this output with one or more input values. Here, for example, routing is treated as a secure multi-party computation problem where the result of the routing algorithm has to be computed using private data held by the candidate nodes to carry messages.

In the passive adversary mode, node A exchanges truthful messages and then analyses them trying to obtain information about the metrics or neighboring graph of node B.

The similarity privacy mechanism protects node A's privacy by hiding its neighboring graph from node B. H_A is first encoded as the coefficients of a polynomial, and then transmitted to B in encrypted form. B cannot decrypt it due to the properties of the Paillier homomorphic encryption scheme but can perform simple operations on the received values. B computes a function that takes A's encrypted values and its own values as inputs and sends the result to A, that decrypts using its private key. A's inputs are hidden because of the multiplication by random once-use numbers. The same plaintext encrypted twice results in two different outputs because the encryption scheme is probabilistic, hence preventing parties from directly comparing the encrypted values. However, node A discloses the number of nodes that are being compared by subtracting the number of coefficients over the number of polynomials. The latter allows node B to decide a minimum number of nodes below which it will not compute the intersection, therefore preventing A from running the mechanisms for a neighboring graph composed of one node only and learning whether or not B's neighboring graph possesses that specific node.

The similarity privacy mechanism achieves security in the semi-honest setting [32]. The latter setting assumes that the parties do not deviate from the protocol. However, the mechanism is not secure against active adversaries. Meanwhile, an active eavesdropper deploying a man-in-the-middle attack, for example, by modifying the messages between the parties, can be detected by adding signatures to the messages, the case of a malicious node A or B may require modifications to the mechanism. Specifically, a zero-knowledge proof protocol can be used to prevent B from pretending it always has a degree of similarity with A. It is assumed here that B has no interest in pretending not to have a similarity with A since in that case, it could just not reply to A's queries.

Some knowledge can be obtained by node A at the end of both link and attribute privacy mechanisms.

The link privacy mechanism protects privacy by obfuscating, that is, by mixing strong with random links and by modifying the weights of links of the neighboring graphs of the nodes. Nevertheless, node A obtains some information about the other nodes that node B came in contact with and vice-versa. According to the mechanism, if two nodes have multiple consecutive encounters, they share the same anonymized information, which will be outdated as at the end of each encounter among two nodes, they update their neighboring graphs accordingly.

For the attribute privacy mechanisms, if node B is found to be more suitable to carry the message, then A infers that B is a better candidate and that $\mathfrak{m}_B$ is higher than $\mathfrak{m}_A$. If B is found to be less suitable, then A infers that $\mathfrak{m}_B$ is lower than $\mathfrak{m}_A$, but neither A nor B knows the order of magnitude in which $\mathfrak{m}_B$ is higher or lower than $\mathfrak{m}_A$. The knowledge that can be learned by A, which carries the message, about $\mathfrak{m}_B$, i.e., the metric of B can be inferred, in all cases, using the output of the protocol and the input provided by A. No information can

be learned from the messages exchanged with B as the ones that A can decrypt are randomized through the use of random nonce values.

The knowledge obtained by B depends on the forwarding policy of A. Recall that B does not know the output of the mechanism. It only knows that the message has finally been forwarded or not to it. If it is considered that the forwarding policy used makes possible not to send the message when B is more suitable, or to send the message even if B is less suitable, then B cannot infer the output of the mechanism. Thus, in this case, B cannot learn anything about $\mathfrak{m}_A$. Next, the worst-case scenario is considered where B knows A's forwarding policy, i.e., a direct single- or multi-copy forwarding policy that allows node B to know the output of the attribute privacy mechanism from the forwarding of the message. If the message is sent, B infers that it is a more suitable candidate than A. If the message is not sent, B learns that it is a less suitable candidate. No information can be learned from the message received from node A because $\mathfrak{m}_A$ is encrypted with A's key.

In summary, all that can be learned by node A about $\mathfrak{m}_B$, or by node B about $\mathfrak{m}_A$, from the attribute privacy mechanism is also learnable from the output alone. Hence, the attribute privacy mechanism protects the privacy of both nodes A and B as it reveals only the result of the algorithm and inferences derived from this result.

B. Active adversary mode

In the active adversary mode, it is considered that an attacker may use untruthful information about its own metric, the messages it carries, or about its neighboring graph, in order to disclose private information about the other part's metrics or neighboring graph.

Node B does not initiate the execution of the attribute privacy mechanism, nor controls the number of messages that will be forwarded. The only chance B has to lie is by manipulating the results of the comparisons sent in Step 3. Node B can lie about its metric, using $\mathfrak{m}_B^{'}$ instead of $\mathfrak{m}_B$, or it can lie about its neighboring graph sending $H_B^{'} = \left(V_B^{'}, E_B^{'}\right)$ instead of $H_B = (V_B, E_B)$ for the link privacy mechanism. Given that using a tampered $H_B^{'}$ will lead to the calculation of an approximate ego betweenness centrality and weight similarity to the destination, the latter is similar to the binary anonymization case. For the attribute privacy mechanism, node B may obtain more information about $\mathfrak{m}_A$ by lying than by being truthful only if it finally receives the message and $\mathfrak{m}_B^{'} > \mathfrak{m}_B$, or if it does not receive the message and $\mathfrak{m}_B^{'} < \mathfrak{m}_B$. In both cases, $\mathfrak{m}_A$ is unknown to B. For these reasons, there is not a straightforward strategy to select $\mathfrak{m}_B^{'}$ that guarantees that B will take an advantage from it.

In summary, an active attacker can try to learn things about the other part's metric by using untruthful information during the execution of the attribute privacy mechanism. Node A can try to learn the metric of node B and vice-versa. In both cases, the attacker obtains the same information that it could infer from a truthful execution of the mechanism.

As A is the one starting the transaction, it is the only one that knows the number of messages that it carries, and to determine how many times to execute the attribute privacy mechanism. If A executes the mechanism enough times, using untruthful information, it may completely uncover $\mathfrak{m}_B$. There is no way for a node to distinguish a truthful execution of the attribute privacy mechanism from an untruthful one given that nodes always operate with encrypted data. However, node B can decrease the effectiveness of these attacks by limiting the number of interactions per unit of time with every other node.

VI. Performance evaluation

This section presents the simulation model and results regarding the performance evaluation of ePRIVO.

A. The simulation model

ePRIVO was implemented in the Opportunistic Network Environment (ONE) simulator [33]. Different simulation scenarios consisting of synthetic mobility models and a real mobility trace were considered. It is assumed here, as in most networks of interest, that there is some social structure between the nodes participating in the network. Each source node generated a new message according to the following intervals: 0.5 to 1 min (0.5-1), 1 to 2 min (1-2), 2 to 4 min (2-4), 4 to 8 min (4-8), and 8 to 16 min (8-16). The length of the timeslots varied from 5, 10, 15, 30 and 60 min corresponding to 288, 144, 96, 48 and 24 timeslots per day, respectively. Similarly to values normally used in the Round Trip Time (RTT) estimation in the Transport Control Protocol (TCP) [34], α and β were set to 0.125 and 0.25, respectively. The weight threshold and number of timeslots were set to 1×10^{-8} and 144 respectively, as explained in [5].

1) Synthetic mobility models: The simulation time was 7 days with an update interval of 1.0 s. Map-based mobility models of Helsinki city over an area of 4.5 × 3.4 Km and Barcelona city over an area of 12 × 12 Km were used. The message size varies from 1 MB to 5 MB. Only two nodes within range can communicate with each other at a time. According to [35], communication in urban environments are highly impaired by obstacles and increasing the transmission range until a certain point saturates the throughput due to higher interference. It was assumed that all nodes used Bluetooth and 802.11a Wi-Fi interfaces. Given that Helsinki and Barcelona cities are urban areas, the communication range between nodes was 10 m and the communication was bidirectional at a constant transmission rate of 2 Mbit/s for the Bluetooth interface. For the Wi-Fi interface, the following communication ranges and transmission rates were considered: 10 m with 10 Mbit/s, and 30 m with 6 Mbit/s, respectively. From time to time, a source node randomly chosen generated one message to a randomly chosen destination. Three mobility models were considered:

a) Shortest-path Map-Based Movement (SPMBM): SPMBM [5] consisted of a network with 40 pedestrians, 80 cars and 6 trams in Helsinki city. Pedestrians were moving at a speed varying between 0.8 to 1.4 m/s. Cars and trams were moving at a speed varying between 2.7 to 13.9 m/s. Each time a tram reaches its destination, it paused for 60 to 300 s. The TTL attribute of each message was 5 h. The pedestrians and

cars had a buffer size of 128 MB. Trams had a buffer size of 512 MB for VDTN traffic.

b) Working Day Movement (WDM): WDM [36] consisted of a network with 110 pedestrians and 32 buses in Helsinki city. There were 50 offices and the working day length was 8 h. The probability of going shopping after work was 50% and there were 10 meeting points. Pedestrians and buses were moving at a speed varying between 0.8 to 1.4 m/s and 7 to 10 m/s, respectively. Each time a bus reaches its destination, it paused for 60 to 300 s. The TTL attribute of each message was 24 h. All nodes had a buffer size of 128 MB for VDTN traffic.

c) Map-Based Movement (MBM): MBM consisted of a network with 90 cars and 6 RSUs in Barcelona city. Cars were moving at a speed varying between 2.7 to 13.9 m/s. Each time a vehicle reaches its destination, it paused for 60 to 300 s. The TTL attribute of each message was 5 h. Cars and RSUs had a buffer size of 128 and 512 MB, respectively. From time to time, a car randomly chosen generated one message to a randomly chosen RSU.

2) Real mobility trace: The taxicabs in Rome (TR) [37] traces, which resembles a set of vehicles across a different network and mobile environment, are used to provide additional support to the analysis and findings of this article. TR contains GPS coordinates of approximately 320 taxicabs collected over 30 days in Rome, Italy. The simulation duration and number of nodes of TR were reduced to 3 days and 304 nodes, respectively. It was assumed that all nodes used an 802.11p Wi-Fi interface with the following communication ranges and transmission rates: 100 m with 10 Mbit/s, and 250 m with 6 Mbit/s, respectively. The message size varies from 50 KB to 500 KB. All nodes had a buffer size of 10 MB for VDTN traffic. The TTL attribute of each message was 24 h.

B. Simulation results

In this section, several simulation results describing the performance of ePRIVO are presented. For each setting, i.e., protocol-configuration parameter pair, at most five independent simulations using different message generation seeds were conducted, and the results averaged, for statistical confidence. ePRIVO was compared with privacy-preserving routing protocols, namely PRIVO and PrivHab+ [22], and with well-known DTN routing protocols [5]: two non-social-based routing protocols, namely Epidemic and Prophet, and two social-based routing protocols, namely BubbleRap and dLife.

The four variants of ePRIVO were considered: ePrivoASP, ePrivoMTTE, ePrivoSDBC and ePrivoCOMBINED.

The performance of ePRIVO was evaluated according to the following metrics: delivery ratio, overhead ratio, and cryptographic cost. The delivery ratio is a key performance indicator as it tells the percentage of successfully received packets of all sent. The overhead ratio is the number of message transmissions for each delivered message. The cryptographic cost, because of homomorphic encryption, gives the computation and transmission cost incurred by cryptographic operations.

The performance of ePRIVO under different forwarding policies will be evaluated. The following forwarding policies were considered: direct single-copy policy (DSCP), direct multi-copy policy (DMCP) and limited multi-copy policy (LMCP).

In addition, information loss (or data utility) due to the use of anonymization methods will also be evaluated. This will be accomplished by analyzing the correlation coefficients between a non-anonymized version of ePRIVO and the anonymized ones over the simulations.

1) The routing performance of ePRIVO with DMCP: This section analyses the routing performance of ePRIVO with the DMCP policy (hereafter ePRIVO*) without the use of homomorphic encryption.

Fig. 3 presents the average delivery ratio and overhead ratio for different scenarios, routing protocols, and message generation rates. As expected, with the decrease of the data rate there is an increase in delivery ratio and a decrease of the overhead ratio as fewer messages circulated in the network.

Overall, ePRIVO* performed better than other routing protocols in all message generation rates and scenarios in terms of delivery and overhead ratios. However, the performance of each ePRIVO* variant depends on the scenario. The routing protocols that presented the highest delivery ratio were ePrivoASP* for TR and ePrivoSDBC* for SPMBM and WDM. The maximum gains obtained for both transmission rates are presented in Table II. Among the non-ePRIVO routing protocols, the ones that presented the highest delivery ratios were Epidemic for SPMBM and WDM, and dLife for TR. Therefore, if there are some repetitive movement patterns then ePrivoSDBC* is the best choice, otherwise, it is ePrivoASP*.

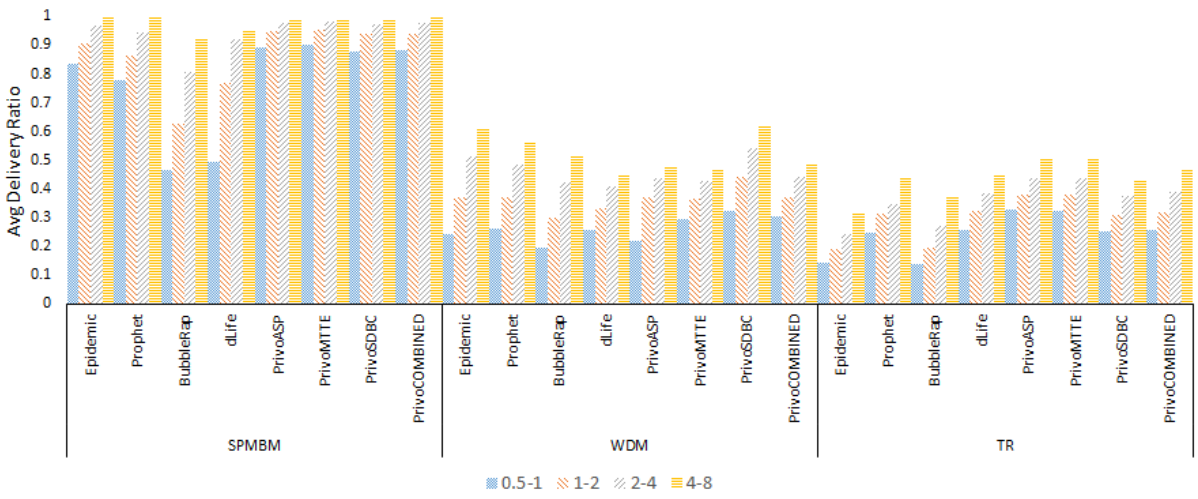
In terms of overhead ratio, Epidemic performed worst due to its unlimited replication approach and was followed by Prophet. For instance, dLife presented the lowest overhead in most cases. ePRIVO* variants presented the second lowest values of overhead ratio. However, they also performed better in terms of the delivery ratio as explained before.

TABLE II
MAXIMUM AVERAGE DELIVERY RATIO GAINS IN DIFFERENT SCENARIOS FOR DIFFERENT MESSAGE GENERATION AND TRANSMISSION RATES (%)

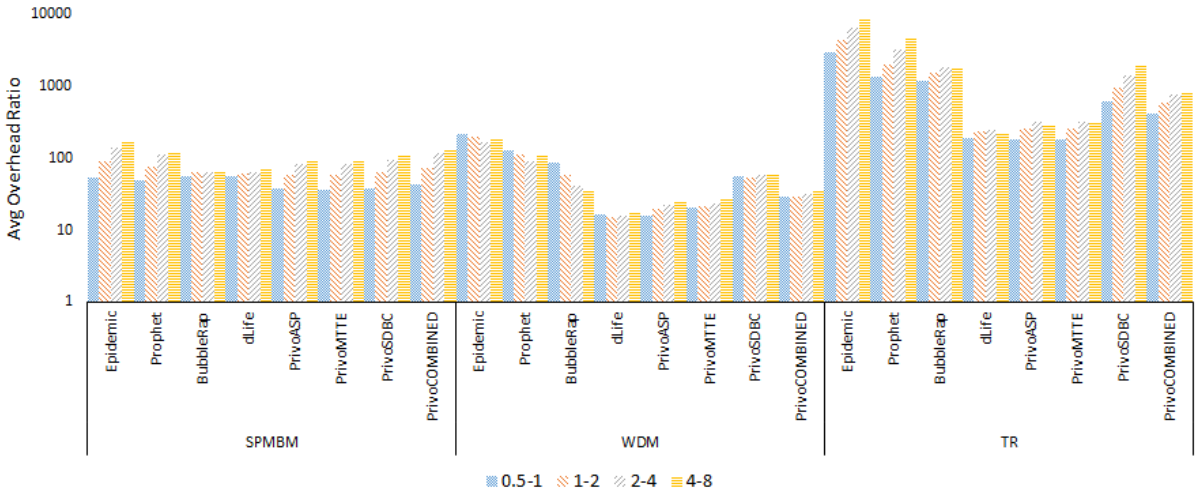
Transmission rate	Mobility model		
	SPMBM	WDM	TR
6 Mbit/s	1.7	37.6	54.6
10 Mbit/s	8.0	37.6	59.8

2) Cryptographic costs: This section analyses the cryptographic cost of using the Paillier homomorphic encryption scheme.

a) Additive homomorphic encryption: A set of experiments were performed to evaluate the performance of additive homomorphic encryption using the Paillier cryptosystem. The experiments were performed on a personal computer with the following specifications: Intel® CORE™ i7-2600 CPU @ 3.40GHz, 16 GB RAM and Windows 10 Pro (64-bits). Table III presents the average Paillier execution time of five operations, namely encryption $\mathcal{E}(a)$, decryption $\mathcal{D}(c)$, sum $\mathcal{E}(a+b)$, difference $\mathcal{E}(a-b)$ and multiplication by a constant $\mathcal{E}(k \cdot a)$. The difference is performed by multiplying



(a) Delivery ratio



(b) Overhead ratio

Fig. 3. Delivery and overhead ratios for all the routing protocols considered in different scenarios for different message generation rates with a transmission rate of 10 Mbit/s.

the second term by -1 followed by summing the numbers, therefore being slower than sum and multiplication by a constant. The operations were repeated 100 times.

b) The performance of ePRIVO with Paillier:* Now, messages were generated every 8 to 16 min. Table IV presents the average delivery ratio losses (+) and gains (-) of ePRIVO* using the Paillier cryptosystem with key sizes of 1024, 2048 and 3072 bits for $\eta = 144$. Each table entry results from averaging losses and gains of all ePRIVO* variants per key. From Table IV, it is possible to see that the losses are below 3% in all scenarios, with exception of WDM, therefore the use of the Paillier homomorphic encryption was not considered in the previous subsection (Section VI-B1).

A more detailed analysis was performed for the legacy key size (i.e., 1024 bits). Table V presents the average delivery ratio losses (+) and gains (-) using the Paillier cryptosystem for $\eta = 144$.

It was concluded, based on simulation results, that if a message was not transmitted because of the additional delay caused by homomorphic encryption, it would be transmitted later on. In some cases (see Table IV and Table V), this additional delay is beneficial to the routing protocol, as it may contribute to the reduction of the network load, even though the maximum achieved gains being negligible (at most 0.50% for the legacy key).

3) Information loss: This section analyses the utility of the data (or information loss) because of the use of anonymization methods. Information loss is measured by comparing the correlation coefficients [38] of the ego betweenness centrality values of all the nodes in the simulation with and without anonymization. The ego betweenness values were collected at the end of each day and the values were compared for different percentages of total anonymization with the case were no anonymization was used. Total anonymization corresponds to the total number of nodes in the neighboring graph that are anonymized. Binary anonymization was applied over a percentage of the latter. At the end of each simulation, the correlation coefficients were averaged taking into account the

TABLE III
AVERAGE PAILLIER EXECUTION TIMES (MS)

Key Size	$\mathcal{E}(a)$	$\mathcal{D}(c)$	$\mathcal{E}(a+b)$	$\mathcal{E}(a-b)$	$\mathcal{E}(k \cdot a)$
1024	11.03 ± 0.1261	11.29 ± 0.3552	0.03 ± 0.0019	0.74 ± 0.0425	0.05 ± 0.0023
2048	83.49 ± 0.3029	83.9 ± 0.4546	0.06 ± 0.0033	1.74 ± 0.0719	0.14 ± 0.0038
3072	271.96 ± 0.123	261.32 ± 0.122	0.08 ± 0.001	3.11 ± 0.007	0.40 ± 0.002

TABLE IV
AVERAGE DELIVERY RATIO LOSSES (+) AND GAINS (-) USING THE PAILLIER CRYPTOSYSTEM (%)

Scenarios	Key size (bits)		
	1024	2048	3072
SPMBM	0.09	0.51	2.58
WDM	4.97	18.28	30.64
TR	-0.12	0.0	0.45

TABLE V
AVERAGE DELIVERY RATIO LOSSES (+) AND GAINS (-) USING THE PAILLIER CRYPTOSYSTEM WITH 1024 BITS KEY (%)

ePRIVO* Variants	Scenarios		
	SPMBM	WDM	TR
ePrivoASP*	0.12	4.72	0.00
ePrivoMTTE*	0.24	4.02	0.00
ePrivoSDBC*	0.00	5.56	0.00
ePrivoCOMBINED*	0.00	5.60	-0.49

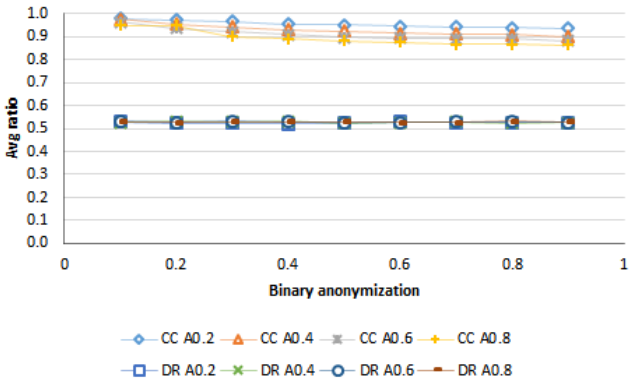


Fig. 4. The average correlation coefficient (CC) and the delivery ratio (DR) for ePrivoSDBC* in the TR scenario for different percentages of total (A) and binary anonymizations

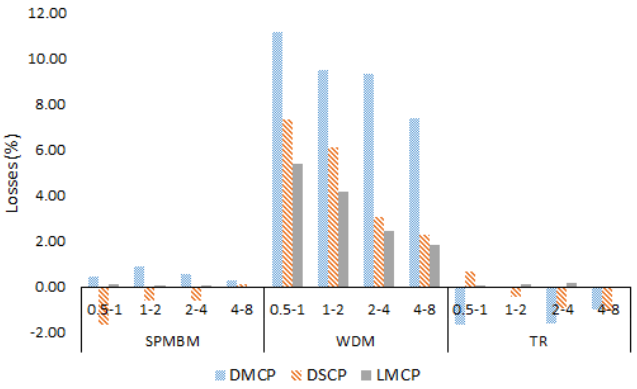


Fig. 5. Average delivery ratio losses (+) and gains (-) of ePRIVO in different scenarios with different message generation rates using the legacy key

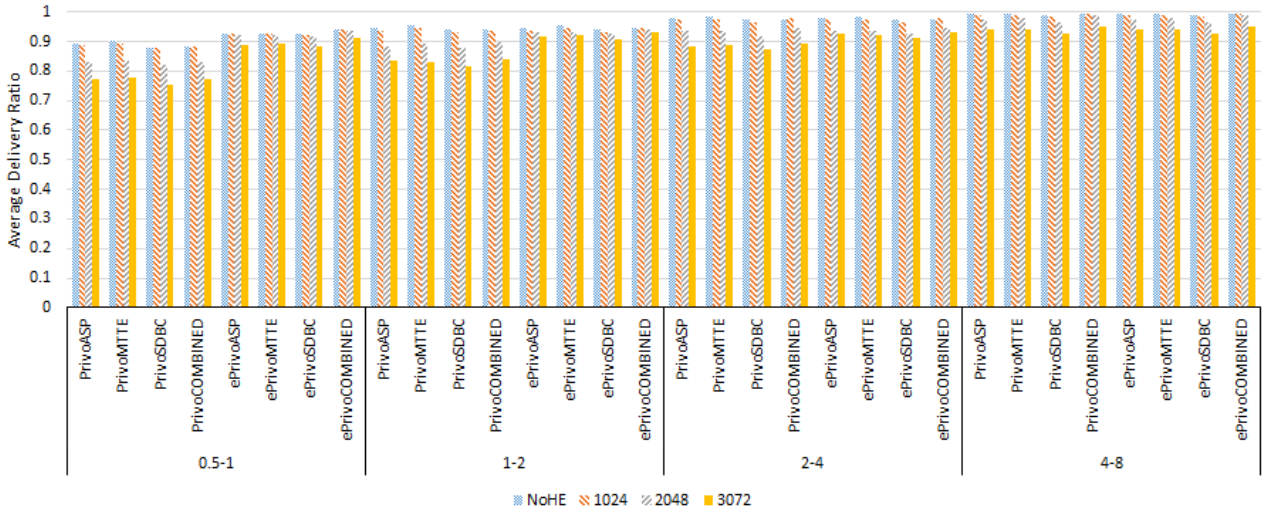
number of days of the simulation. Different percentages of binary and total anonymization were used. The former varied from 10% to 90% with increments of 10% and the latter varied from 20% to 80% with increments of 20%.

Fig. 4 presents the average correlation coefficient (CC) and the delivery ratio (DR) for ePrivoSDBC* in the TR scenario. Between binary anonymization and neighborhood randomization, the former is the one to cause a reduction on the average correlation coefficients as it increases, and this effect worsens as the percentage of total anonymization increases. Nonetheless, since ePrivoSDBC* uses ego betweenness centrality and weighted similarity to the destination and the latter is more frequently used as a routing metric, the effects of the lowest values of correlation coefficients (i.e., 0.86 for TR corresponding to 90% of binary anonymization and 80% of total anonymization) are not significant as can be seen by the steady average delivery ratio in Fig. 4.

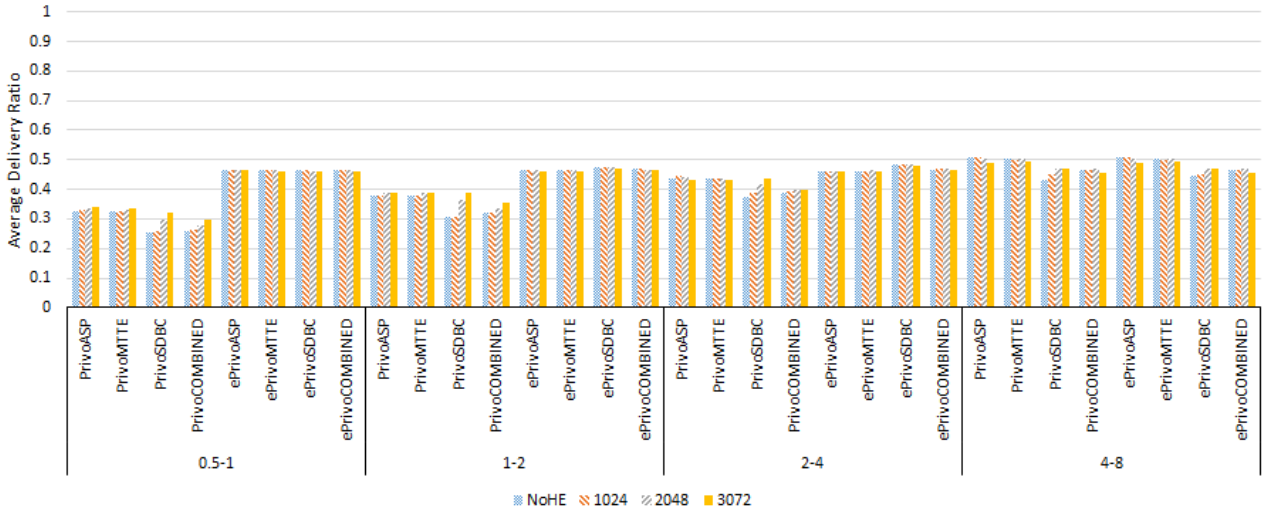
4) ePRIVO's forwarding policies: This section analyses the routing performance of ePRIVO with different forwarding policies. Fig. 5 presents the average delivery ratio losses (+) and gains (-) of ePRIVO in SPMBM, WDM and TR scenarios for DSCP, DMCP and LMCP forwarding policies for different message generation rates using the legacy key as compared with the scenario without key.

In general, the reduction of the message generation rates caused a reduction of the average losses and gains of different forwarding policies for SPMBM, WDM and TR scenarios, respectively. At one side is DMCP that presents the highest losses and gains due to the excessive number of messages generated by the policy. Please note that despite the losses, DMCP is the best forwarding policy for the WDM scenario because of the low opportunity of contacts among nodes that is characteristic of the scenario. In SPMBM and TR, DMCP presented the highest gains as the forwarding policy was the most affected by the increase of the message generation rates. Recall that the node density in TR is higher if compared to SPMBM and WDM and the nodes did not follow predefined routines. On the other hand, LMCP presented the lowest losses in comparison to DMCP and DSCP because of the controlled replication strategy that was beneficial in most cases.

5) ePRIVO's routing performance: This section analyses the routing performance of ePRIVO in comparison with PRIVO and PrivHab+. ePRIVO uses a combination of different forwarding policies that depend on, for example, the scenario



(a) SPMBM



(b) TR

Fig. 6. Comparison of delivery ratio between ePRIVO and PRIVO for SPMBM and TR scenarios with different message generation rates and key sizes

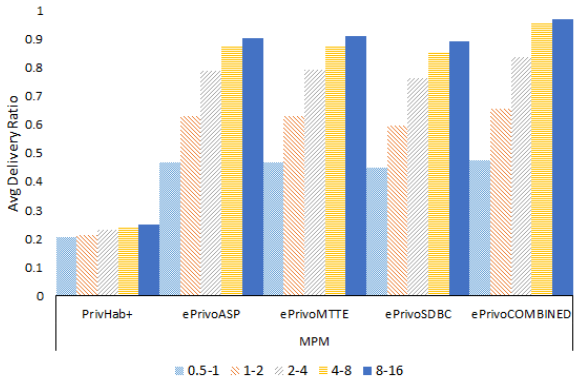
considered, the message generation rates, among others.

Fig. 6 presents a comparison of the delivery ratio between ePRIVO and PRIVO for SPMBM and TR scenarios with different message generation rates and key sizes.

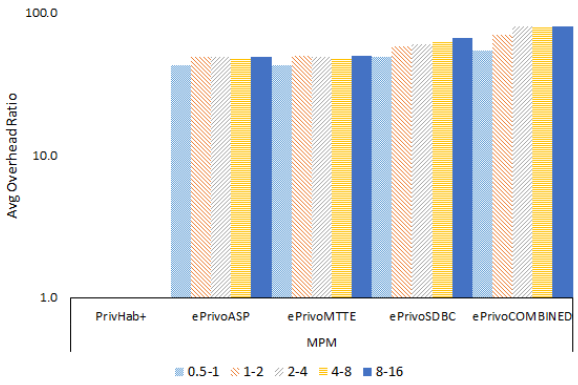
In WDM, both protocols behaved similarly as most of the time they were using the same forwarding policy, that is, both protocols used DMCP. The latter was the main reason why the WDM graph in Figure Fig. 6 was omitted. The main difference between PRIVO and ePRIVO was that for the lowest message generation rate (i.e., 4-8min), ePRIVO also used LMCP that allowed it to obtain a slightly higher delivery ratio. According to Table VI, the average delivery ratio gains of ePRIVO in comparison to PRIVO for WDM were negligible, being at most 0.47% for the 3072 bits key and mainly for the lowest message generation rates.

In SPMBM and TR, ePRIVO outperformed PRIVO because it used a combination of forwarding policies. Specifically, ePRIVO used LMCP for the high message generation rates and DMCP for low message generation rates. According to Table VI, the average delivery ratio gains were 33.12, 27.57 and 23.98% for 1024, 2048 and 3072 bits of key sizes for TR, respectively.

Fig. 7 presents a comparison of the delivery and overhead ratios between ePRIVO and PrivHab+ for the MBM scenario with different message generation rates. MBM consisted of RSUs that were static and vehicles that were moving. Hence, the goal here was for vehicles to forward messages using V2V communications until they reach the RSUs. Please recall that PrivHab+ is a geographical-based privacy-preserving routing protocol aiming at pushing the message as close as possible to the destination node's position. Simulations results showed that ePRIVO outperformed PrivHab+ in terms of the delivery ratio by presenting maximum delivery ratio gains of 288%. However, and since ePRIVO's aim was to maximize the delivery ratio, it also presented higher overhead ratio in comparison to PrivHab+ but smaller than the one presented in Figure 3(b).



(a) Delivery ratio



(b) Overhead ratio

Fig. 7. Comparison of delivery and overhead ratios between ePRIVO and PrivHab+ for the MBM scenario with different message generation rates and a transmission rate of 6 Mbit/s

ePRIVO could limit the number of copies that circulate on the network by means of the LMCP forwarding policy if the objective was to reduce the overhead ratio. By doing so, a maximum reduction of 91.3% would be attained.

In summary, there is not a one-solution-fits-all in terms of forwarding policies since the best results mostly depend on a combination of forwarding policies.

TABLE VI
AVERAGE EPRIVO'S DELIVERY RATIO GAINS AS COMPARED WITH PRIVO (%)

Scenarios	Key size (bits)			
	0	1024	2048	3072
SPMBM	1.92	2.52	5.77	9.27
WDM	0.0	0.0	0.0	0.47
TR	31.73	33.12	27.57	23.98

VII. CONCLUSIONS AND FUTURE WORK

This article proposed ePRIVO, an enhanced PRIvacy-preserVing Opportunistic routing protocol for Vehicular Delay-Tolerant Networks. ePRIVO ensures link privacy by means of binary anonymization and neighborhood randomization, and attribute privacy by means of the Paillier homomorphic encryption scheme. In addition, ePRIVO proposed the similarity privacy mechanism that allowed nodes to calculate their similarity in a private manner.

The effectiveness of ePRIVO is supported through extensive simulations with synthetic mobility models and a real mobility trace. Simulation results show that ePRIVO presents on average very low cryptographic costs in most scenarios, and if there are some repetitive movement patterns then ePrivoSDBC is the best choice, otherwise, it is ePrivoASP. Furthermore, ePRIVO presents on average gains of approximately 4.87 and 29.1% in terms of the delivery ratio for SPMBM and TR scenario compared to PRIVO, respectively. In addition, it also presents average maximum gains of 238.4% in terms of the delivery ratio for the MBM scenario compared to PrivHab+.

As future work, the following research challenges have been identified: (i) the evaluation of ePRIVO with a synthetic mobility model and real mobility trace composed of a very high number of vehicles and (ii) testing ePRIVO in a real testbed.

ACKNOWLEDGMENT

A warm thanks to Zhengguo Sheng. This research was partially supported by Fundação Calouste Gulbenkian and by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UID/CEC/50021/2013. It was also sponsored by The Engineering, and Physical Sciences Research Council (EPSRC) (EP/P025862/1), Royal Society-Newton Mobility Grant (IE160920).

REFERENCES

[1] P. R. Pereira, A. Casaca, J. J. P. C. Rodrigues, V. N. G. J. Soares, J. Triay, and C. Cervello-Pastor, "From delay-tolerant networks to vehicular delay-tolerant networks," *IEEE Communications Surveys Tutorials*, vol. 14, no. 4, pp. 1166–1182, 2012.

[2] M. S. Obaidat and S. Misra, *Cooperative Networking*. Wiley, 2011.

[3] K. Wei, X. Liang, and K. Xu, "A survey of social-aware routing protocols in delay tolerant networks: applications, taxonomy and design-related issues," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 556–578, Jan. 2014.

[4] N. Magaia, A. P. Francisco, P. Pereira, and M. Correia, "Betweenness centrality in delay tolerant networks: A survey," *Ad Hoc Networks*, vol. 33, pp. 284 – 305, 2015.

[5] N. Magaia, C. Borrego, P. Pereira, and M. Correia, "PRIVO: A PRIvacy-preserVing Opportunistic routing protocol for Delay-Tolerant Networks," in *IFIP Networking 2017*, Jun 2017.

[6] N. Magaia, P. R. Pereira, and M. P. Correia, *Cyber Physical Systems: From Theory to Practice*. CRC Press, 2015, ch. Security in Delay-Tolerant Mobile Cyber-Physical Applications.

[7] X. Wu, X. Ying, K. Liu, and L. Chen, "A Survey of Algorithms for Privacy-Preservation of Graphs and Social Networks," *Managing and Mining Graph Data*, p. 37, 2009.

[8] X. Lu, P. Hui, D. Towsley, J. Pu, and Z. Xiong, "Anti-localization anonymous routing for Delay Tolerant Network," *Computer Networks*, vol. 54, no. 11, pp. 1899–1910, 2010.

[9] A. Shikfa, M. Onen, and R. Molva, "Privacy and confidentiality in context-based and epidemic forwarding," *Computer Communications*, vol. 33, no. 13, pp. 1493–1504, 2010.

[10] G. Costantino, F. Martinelli, and P. Santi, "Privacy-preserving interest-casting in opportunistic networks," in *IEEE Wireless Communications and Networking Conference, WCNC*. IEEE, Apr 2012, pp. 2829–2834.

[11] C. Dunbar and G. Qu, "A DTN Routing Protocol for Vehicle Location Information Protection," in *2014 IEEE Military Communications Conference*, Oct 2014, pp. 94–100.

[12] C. P. A. Ogah, H. Cruickshank, Z. Sun, G. Chandrasekaran, Y. Cao, P. M. Asuquo, and M. A. Tawqi, "Privacy-Enhanced Group Communication for Vehicular Delay Tolerant Networks," in *2015 9th International Conference on Next Generation Mobile Applications, Services and Technologies*, Sept 2015, pp. 193–198.

[13] N. Ahmad, H. Cruickshank, Y. Cao, F. A. Khan, M. Asif, A. Ahmad, and G. Jeon, "Privacy by architecture pseudonym framework for delay tolerant network," *Future Generation Computer Systems*, 2017.
[14] Y. Park, C. Sur, and K.-H. Rhee, "A simplified privacy preserving message delivery protocol in VDTNs," in *Information and Communication Technology*, K. Mustofa, E. J. Neuhold, A. M. Tjoa, E. Weippl, and I. You, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 416–425.
[15] M. Radenkovic and I. Vaghi, "Adaptive user anonymity for mobile opportunistic networks," in *Proceedings of the Seventh ACM International Workshop on Challenged Networks*, ser. CHANTS '12. New York, NY, USA: ACM, 2012, pp. 79–82.
[16] I. Parris and T. Henderson, "Privacy-enhanced social-network routing," *Computer Communications*, vol. 35, no. 1, pp. 62–74, 2012.
[17] K. Chen and H. Shen, "Distributed Privacy-Protecting Routing in DTN: Concealing the Information Indispensable in Routing," in *2017 14th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, June 2017, pp. 1–9.
[18] J. Zhou, X. Dong, Z. Cao, and A. V. Vasilakos, "Secure and Privacy Preserving Protocol for Cloud-Based Vehicular DTNs," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 6, pp. 1299–1314, June 2015.
[19] J. Miao, O. Hasan, S. B. Mokhtar, L. Brunie, and A. Hasan, "4pr: Privacy preserving routing in mobile delay tolerant networks," *Computer Networks*, vol. 111, pp. 17 – 28, 2016, cyber-physical systems for Mobile Opportunistic Networking in Proximity (MNP).
[20] R. L. Rivest, L. Adleman, and M. L. Dertouzos, "On data banks and privacy homomorphims," *Foundations of secure computation*, vol. 4, no. 11, pp. 169–180, 1978.
[21] N. Zeilemaker, Z. Erkin, P. Palmieri, and J. Pouwelse, "Building a privacy-preserving semantic overlay for peer-to-peer networks," in *Information Forensics and Security (WIFS), 2013 IEEE International Workshop on*, Nov 2013, pp. 79–84.
[22] A. Sánchez-Carmona, S. Robles, and C. Borrego, "PrivHab+: A secure geographic routing protocol for DTN," *Computer Communications*, vol. 78, pp. 56–73, 2016.
[23] P. Paillier, "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes," in *Advances in Cryptology — EUROCRYPT '99*, ser. Lecture Notes in Computer Science, J. Stern, Ed. Springer Berlin Heidelberg, 1999, vol. 1592, pp. 223–238.
[24] Y. Li, D. Jin, P. Hui, and S. Chen, "Contact-Aware Data Replication in Roadside Unit Aided Vehicular Delay Tolerant Networks," *IEEE Transactions on Mobile Computing*, vol. 15, no. 2, pp. 306–321, Feb 2016.
[25] M. Everett and S. P. Borgatti, "Ego network betweenness," *Social Networks*, vol. 27, no. 1, pp. 31–38, Jan. 2005.
[26] E. M. Daly and M. Haahr, "Social Network Analysis for Routing in Disconnected Delay-tolerant MANETs," in *Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '07. New York, NY, USA: ACM, 2007, pp. 32–40.
[27] M. Mcpherson, L. Smith-Lovin, and J. M. Cook, "Birds of a Feather: Homophily in Social Networks," *Annual Review of Sociology*, vol. 27, no. 1, pp. 415–444, 2001.
[28] M. J. Freedman, K. Nissim, and B. Pinkas, "Efficient Private Matching and Set Intersection," in *Advances in Cryptology - EUROCRYPT 2004*, ser. Lecture Notes in Computer Science, C. Cachin and J. Camenisch, Eds. Springer Berlin Heidelberg, 2004, vol. 3027, pp. 1–19.
[29] G. Zhong, I. Goldberg, and U. Hengartner, "Louis, Lester and Pierre: Three Protocols for Location Privacy," *Pets'07*, pp. 62–76, 2007.
[30] O. Goldreich, "Secure multi-party computation," *Manuscript. Preliminary version*, pp. 86–97, 1998.
[31] M. J. Atallah and M. Blanton, *Algorithms and theory of computation handbook, volume 2: special topics and techniques*. Chapman and Hall/CRC, 2009.
[32] M. J. Freedman, K. Nissim, and B. Pinkas, *Efficient Private Matching and Set Intersection*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 1–19.
[33] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE simulator for DTN protocol evaluation," in *Proceedings of the 2nd international conference on simulation tools and techniques*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009, p. 55.
[34] J. Kurose and K. Ross, *Computer Networking - A top-down approach*. Pearson, 2012.
[35] J. S. Otto, F. E. Bustamante, and R. A. Berry, "Down the Block and Around the Corner The Impact of Radio Propagation on Inter-vehicle Wireless Communication," in *2009 29th IEEE International Conference on Distributed Computing Systems*, June 2009, pp. 605–614.
[36] F. Ekman, A. Keränen, J. Karvo, and J. Ott, "Working Day Movement Model," in *Proceedings of the 1st ACM SIGMOBILE Workshop on Mobility Models*, ser. MobilityModels '08. New York, NY, USA: ACM, 2008, pp. 33–40.
[37] L. Bracciale, M. Bonola, P. Loreti, G. Bianchi, R. Amici, and A. Rabuffi, "CRAWDAD dataset roma/taxi (v. 2014-07-17)," Downloaded from http://crawdad.org/roma/taxi/20140717, Jul. 2014.
[38] R. A. Fisher, "Frequency distribution of the values of the correlation coefficient in samples from an indefinitely large population," *Biometrika*, vol. 10, no. 4, pp. 507–521, May 1915.

Naercio Magaia received his PhD with distinction in Electrical and Computer Engineering at Instituto Superior Técnico (IST), Universidade de Lisboa (ULisboa). He holds a degree in Electrical Engineering from Eduardo Mondlane University, and an M.Sc. in Communication Networks Enginnering from IST, ULisboa. His current research interests cover vehicular communication, delay-tolerant networks, network security, edge computing and multi-objective optimization.

Carlos Borrego received his degree in Computer Science (6 year programme) at the Faculty of Computer Science at the Polytechnic University of Madrid. After finishing his studies he moved to work for CERN (Geneva, Switzerland). In 2001 moved to CASPUR, University La Sapienza (Rome, Italy) and stayed there for four years. In 2005 moved to the Autonomous University of Barcelona (Barcelona, Spain) where he finished his PhD and worked for Pic and Ifae research centers. He is actually researcher and adjunct professor at the Department of Information and Communications Engineering dEIC. He gives lectures on computer networks and cryptography.

Paulo Rogério Pereira (S'97, M'04, SM'15) received his Ph.D. in Electrical and Computer Science Engineering from Instituto Superior Técnico, University of Lisbon (IST/UL), Portugal, in 2003. He is an assistant professor of computer networks at IST/UL and a senior researcher at INESC-ID. He has participated in the IST European projects EuroNGI, EuroFGI, EuroNF, UbiSec&Sens, WSAN4CIP and E-Balance. His research interests include IP wireless sensor networks, delay-tolerant networks, quality of service and network management.

Miguel Correia is an Associate Professor at Instituto Superior Técnico (IST) of Universidade de Lisboa (ULisboa and Senior Researcher at INESC-ID in Lisboa, Portugal. He has been involved in many research projects related to cybersecurity including the SafeCloud, PCAS, TCLOUDS, ReSIST, CRUTIAL, and MAFTIA European projects. He has more than 150 publications. His research is focused on (cyber)security and dependability (aka fault tolerance), typically in distributed systems, in the context of different applications (blockchain, cloud, mobile).